**[ 10 ] 자바빈**

1. 빈이란?
   * 반복적인 작업을 효율적으로 하기 위해 빈을 사용한다. 빈이란? JAVA언어의 데이터(속성)와 기능(메소드)으로 이루어진 클래스이다.
   * jsp페이지를 만들고, 액션태그를 이용하여 빈을 사용한다. 그리고 빈의 내부 데이터를 처리한다
2. 빈 만들기
   * JAVA언어를 학습 하면서 데이터 객체를 많이 만들어본 경험이 있을 것이다. 데이터 객체에는 데이터가 있어 그에 해당하는 getter와 setter가 있다.
   * 빈을 만든다는 것은 데이터 객체를 만들기 위한 클래스를 만드는 것이다.

**public** **class** Student {

**private** **String** studentNum;

**private** **String** name;

**private** **int** grade;

**private** **char** className;

**private** **int** score;

}

1. 빈 관련 액션태그(useBean, getProperty, setProperty) = 빈을 사용하는 방법
   * 액션태그 중에서 Bean관련 태그가 있다. 주로 데이터 업데이트하고 얻어오는 역할을 한다.
   * useBean : 특정 Bean을 사용한다고 명시할 때 사용

<jsp:useBean id="student" class="com.tj.ex.Student" scope="page" />

객체이름 클래스 이름 사용범위

자바에서처럼 만들어진 클래스를 다음과 같은 방법으로 이용한다.

Student student = new Student();

여기서 student가 자바빈에서는 id이다. 자바빈의 이름이 student이다.

* + scope
    1. page : 생성된 페이지 내에서만 사용 가능
    2. request : 요청된 페이지 내에서만 사용 가능
    3. session : 웹브라우저의 생명주기와 동일하게 사용 가능
    4. application : 웹 어플리케이션 생명주기와 동일하게 사용 가능
  + setProperty : 데이터 값을 설정할 때 사용. 자바에서의 setter역할

<jsp:setProperty name="student" property="name" value="홍길동"/>

* + getProperty : 데이터 값을 가져올 때 사용. 자바에서의 getter 역할

<jsp:getProperty name="student" property="name" />

* JSP에서 데이터 객체 사용 ※

Input.html

<!DOCTYPE html>

<html>

<head>

<meta charset=*"UTF-8"*>

<title>Insert title here</title>

</head>

<body>

<!-- <form action="step4.jsp"> -->

<!-- <form action="step5.jsp"> -->

<form action=*"step1.jsp"*>

이름 <input type=*"text"* name=*"name"* required=*"required"*><br>

나이 <input type=*"number"* name=*"age"* required=*"required"*><br>

성별 <input type=*"text"* name=*"gender"* required=*"required"*><br>

주소 <input type=*"text"* name=*"address"*><br>

<input type=*"submit"* value=*"출력"*>

</form>

</body>

</html>

1단계 :

**public** **class** PersonInfo {

**private** String name;

**private** **int** age;

**private** String gender;

**public** String getName() {**return** name;}

**public** **void** setName(String name) {**this**.name = name;}

**public** **int** getAge() {**return** age;}

**public** **void** setAge(**int** age) {**this**.age = age;}

**public** String getGender() {**return** gender;}

**public** **void** setGender(String gender) {**this**.gender = gender;}

}

<%@page import="person.PersonInfo"%>

<body>

<%

PersonInfo pi = **new** PersonInfo();

pi.setName("홍길동");

pi.setAge(23);

pi.setGender("남자");

String name = pi.getName();

**int** age = pi.getAge();

String gender = pi.getGender();

%>

<h2>개인정보</h2>

이 름 : <%=name %><p>

나 이 : <%=age %><p>

성 별 : <%=gender %><p>

<hr>

이 름 : <%=pi.getName()%><p>

나 이 : <%=pi.getAge() %><p>

성 별 : <%=pi.getGender() %><p>

</body>

2단계 :

<body>

<jsp:useBean id=*"pi"* class=*"person.PersonInfo"*></jsp:useBean>

<jsp:setProperty property=*"name"* name=*"pi"* value=*"홍길동"*/> <!-- setName() 찾아감 가능한 요로코롬 자바코드 안 쓰는 것 추천-->

<jsp:setProperty property=*"age"* name=*"pi"* value=*'23'*/>

<jsp:setProperty property=*"gender"* name=*"pi"* value=*"남자"*/>

<h2>개인정보</h2>

<hr>

이 름 : <jsp:getProperty property=*"name"* name=*"pi"*/><p> <!-- getName() 찾아감 -->

나 이 : <jsp:getProperty property=*"age"* name=*"pi"*/><p>

성 별 : <jsp:getProperty property=*"gender"* name=*"pi"*/><p>

<hr>

</body>

3단계 : 뷰와 프로세스를 분리

Person3.jsp

<!--

person3.jsp : 객체 생성 pi = new person/PersonInfo(); scope:request

setProperty : pi 값을 저장

jsp → pi(객체생성 javaBean)

personResult.jsp : request pi 가져와

jsp → getProperty

-->

<%@ page language=*"java"* contentType=*"text/html; charset=UTF-8"*

pageEncoding=*"UTF-8"*%>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

<html>

<head>

<meta http-equiv=*"Content-Type"* content=*"text/html; charset=UTF-8"*>

<title>Insert title here</title>

</head>

<body>

<jsp:useBean id=*"pi"* class=*"person.PersonInfo"* scope=*"request"*/>

<jsp:setProperty property=*"name"* name=*"pi"* value=*"홍길동"*/> <!-- setName() 찾아감 가능한 요로코롬 자바코드 안 쓰는 것 추천-->

<jsp:setProperty property=*"age"* name=*"pi"* value=*'23'*/>

<jsp:setProperty property=*"gender"* name=*"pi"* value=*"남자"*/>

<jsp:forward page=*"personResult.jsp"/*> <!—response.sendRedirect()와 비교-->

</body>

</html>

personResult.jsp

<body>

<jsp:useBean id=*"pi"* class=*"person.PersonInfo"* scope=*"request"*/>

<h2>개인정보</h2>

<hr>

이 름 : <jsp:getProperty property=*"name"* name=*"pi"*/><p>

나 이 : <jsp:getProperty property=*"age"* name=*"pi"*/><p>

성 별 : <jsp:getProperty property=*"gender"* name=*"pi"*/><p>

<hr>

</body>

4단계 : html 폼태그로부터 받은 데이터를 자바빈에 넣는다

Person.html

<body>

<h1>개인정보 입력창</h1>

<!-- <form action="person4.jsp"> -->

<!-- <form action="person5.jsp"> -->

<form action=*"person6.jsp"*>

이름 : <input type=*"text"* name=*"name"* required=*"required"*><p>

나이 : <input type=*"number"* name=*"age"* required=*"required"*><p>

성별 : <label for=*"m1"*>남자</label>

<input type=*"radio"* name=*"gender"* id=*"m1"* value=*"남자"* checked=*"checked"*>

<label for=*"m2"*>여자</label>

<input type=*"radio"* name=*"gender"* id=*"m2"* value=*"여자"* ><p>

<input type=*"submit"* value=*"확인"*>

</form>

</body>

Person4.jsp

<body>

<jsp:useBean id=*"pi"* class=*"person.PersonInfo"* scope=*"request"*/>

<jsp:setProperty property=*"name"* name=*"pi"* value=*'*<%=request.getParameter("name") %>*'*/>

<jsp:setProperty property=*"age"* name=*"pi"* value=*'*<%=Integer.parseInt(request.getParameter("age")) %>*'*/>

<jsp:setProperty property=*"gender"* name=*"pi"* value=*"*<%=request.getParameter(\"gender\")%>*"*/>

<jsp:forward page=*"personResult.jsp"*></jsp:forward>

</body>

5단계 : 파라미터를 바로 집어 넣는다.

<body>

<jsp:useBean id=*"pi"* class=*"person.PersonInfo"* scope=*"request"*/>

<jsp:setProperty property=*"name"* name=*"pi"* param=*"name"*/>

<jsp:setProperty property=*"age"* name=*"pi"* param=*"age"*/>

<jsp:setProperty property=*"gender"* name=*"pi"* param=*"gender"*/>

<jsp:forward page=*"personResult.jsp"*></jsp:forward>

</body>

6단계 : html form안의 이름들과 자바빈의 속성이름이 같으면 아래와 같이 간략화 된다.

<body>

<jsp:useBean id=*"pi"* class=*"person.PersonInfo"* scope=*"request"*/>

<jsp:setProperty property=*"\*"* name=*"pi"*/>

<!-- person.html의 name과 자바빈 PersonInfo의 name이 같아야 요렇게 가능 -->

<!-- person.html의 name과 자바빈 멤버 이름이 다르면 param을 쓴다 -->

<jsp:forward page=*"personResult.jsp"*/>

</body>

연습문제 :

input.html - 사용자로부터 Student 정보 데이터 받기

step4.jsp, step5.jsp, step6.jsp(프로세스단) - input.html에서 받은 파라미터를 st빈의 속성에 넣고

뷰단으로 보내기

stResult.jsp(뷰단) - 프로세스단에서 넣은 값을 보여준다

위의 5개 파일(input.html, st4.jsp, st5.jsp, st6.jsp, stResult.jsp)을 갖고 계세요.